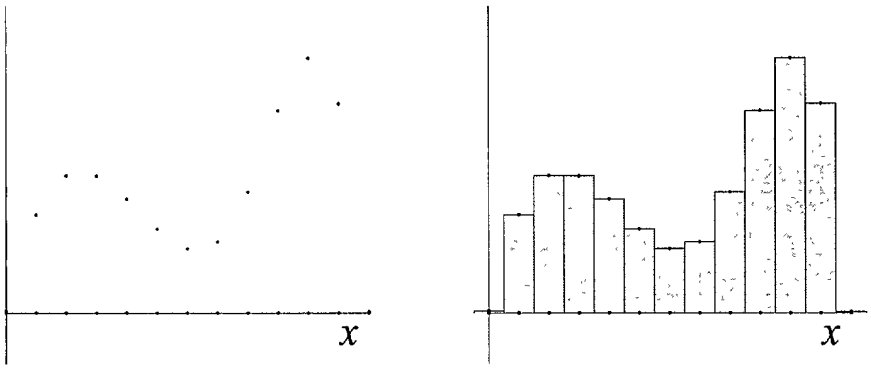
FIGURE 1



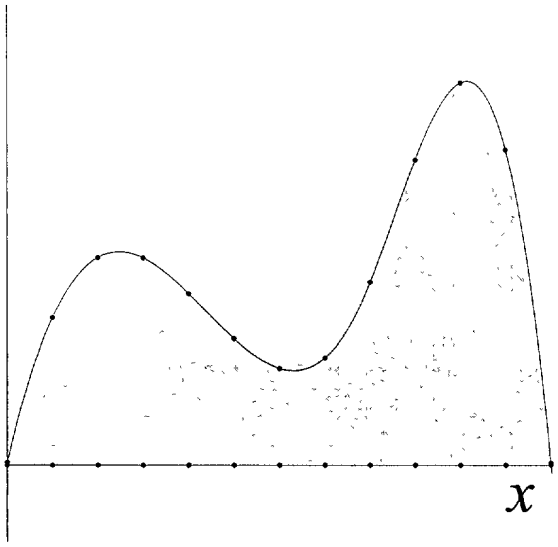


FIGURE 2

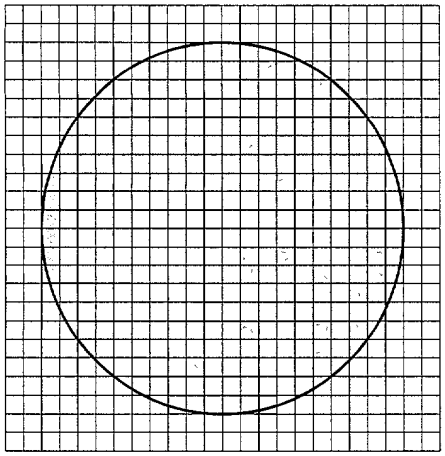


FIGURE 3

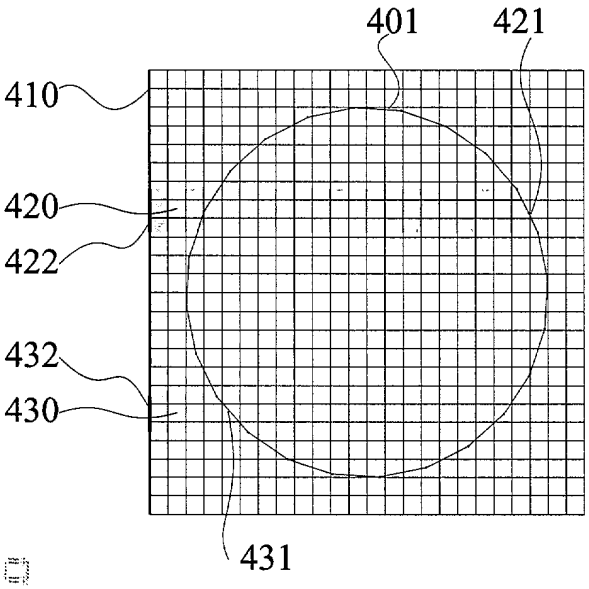
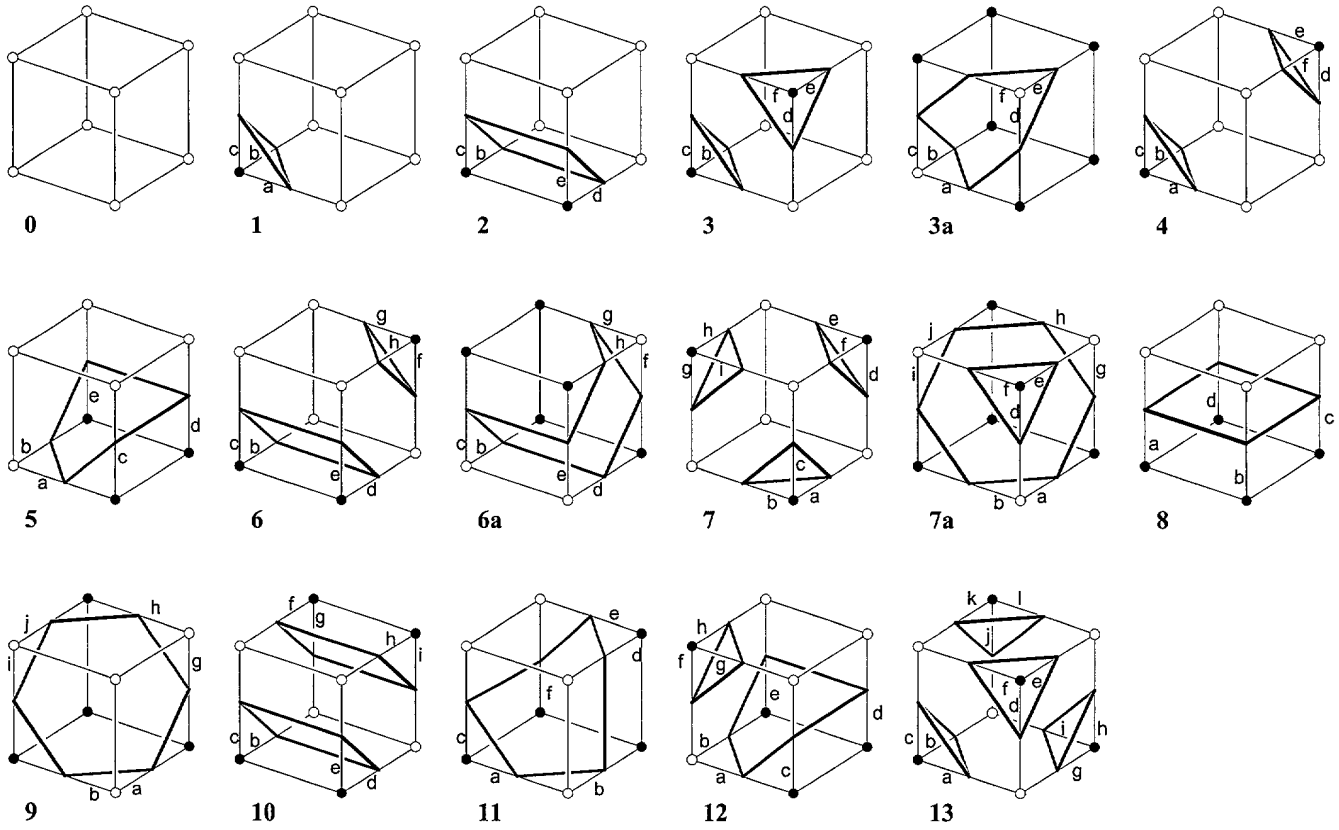


FIGURE 4

FIGURE 5



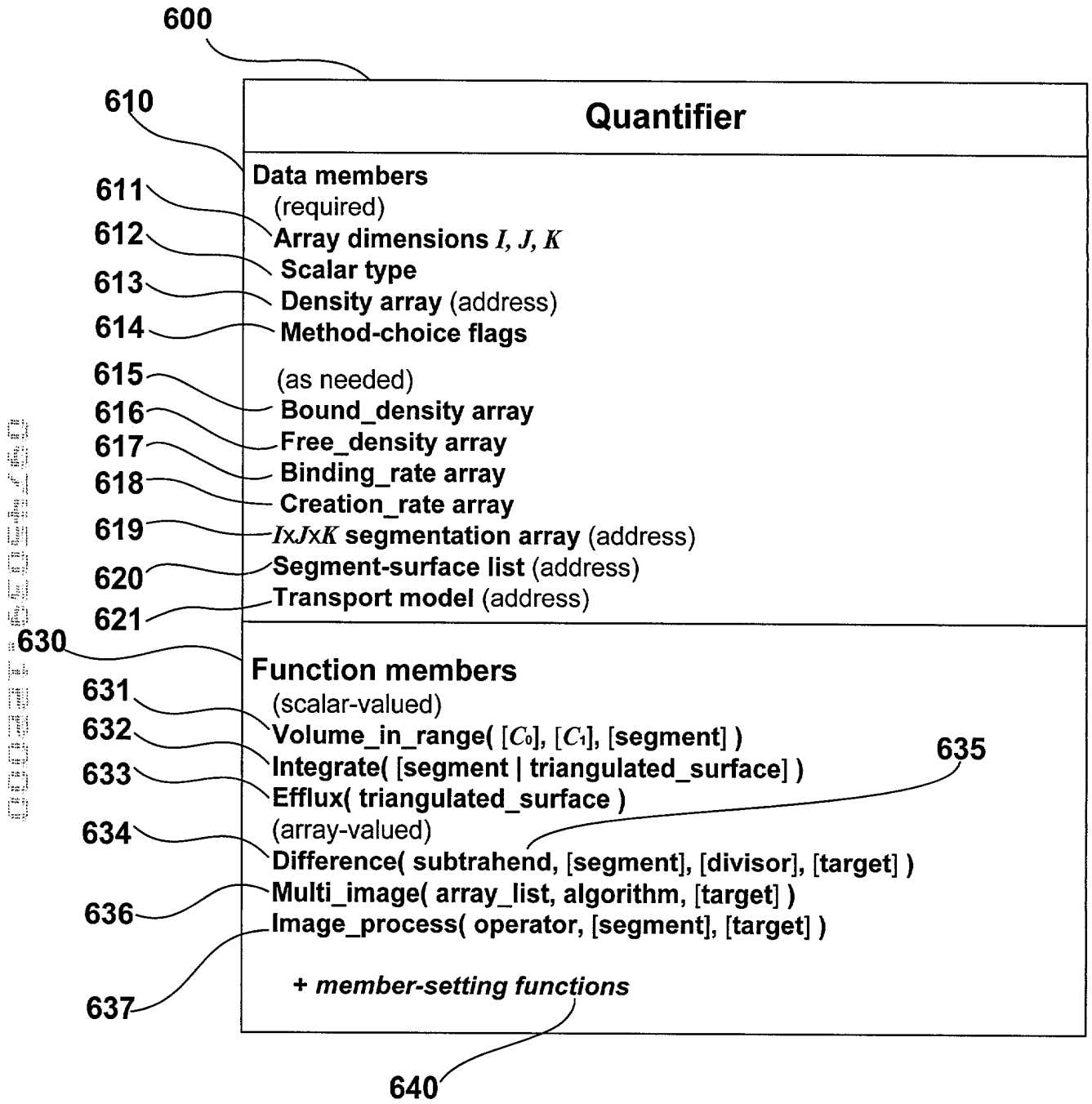

600
610
Quantifier
Data members
(required)
611
Array dimensions I, J, K
612
Scalar type
613
Density array (address)
614
Method-choice flags
(as needed)
615
Bound_density array
616
Free_density array
617
Binding_rate array
618
Creation_rate array
619
IxJxK segmentation array (address)
620
Segment-surface list (address)
621
Transport model (address)
630
Function members
(scalar-valued)
631
Volume_in_range([C0], [C1], [segment])
632
Integrate([segment | triangulated_surface])
633
Efflux(triangulated_surface)
(array-valued)
634
Difference(subtrahend, [segment], [divisor], [target])
635
636
Multi_image(array_list, algorithm, [target])
637
Image_process(operator, [segment], [target])
+ member-setting functions
640

FIGURE 6

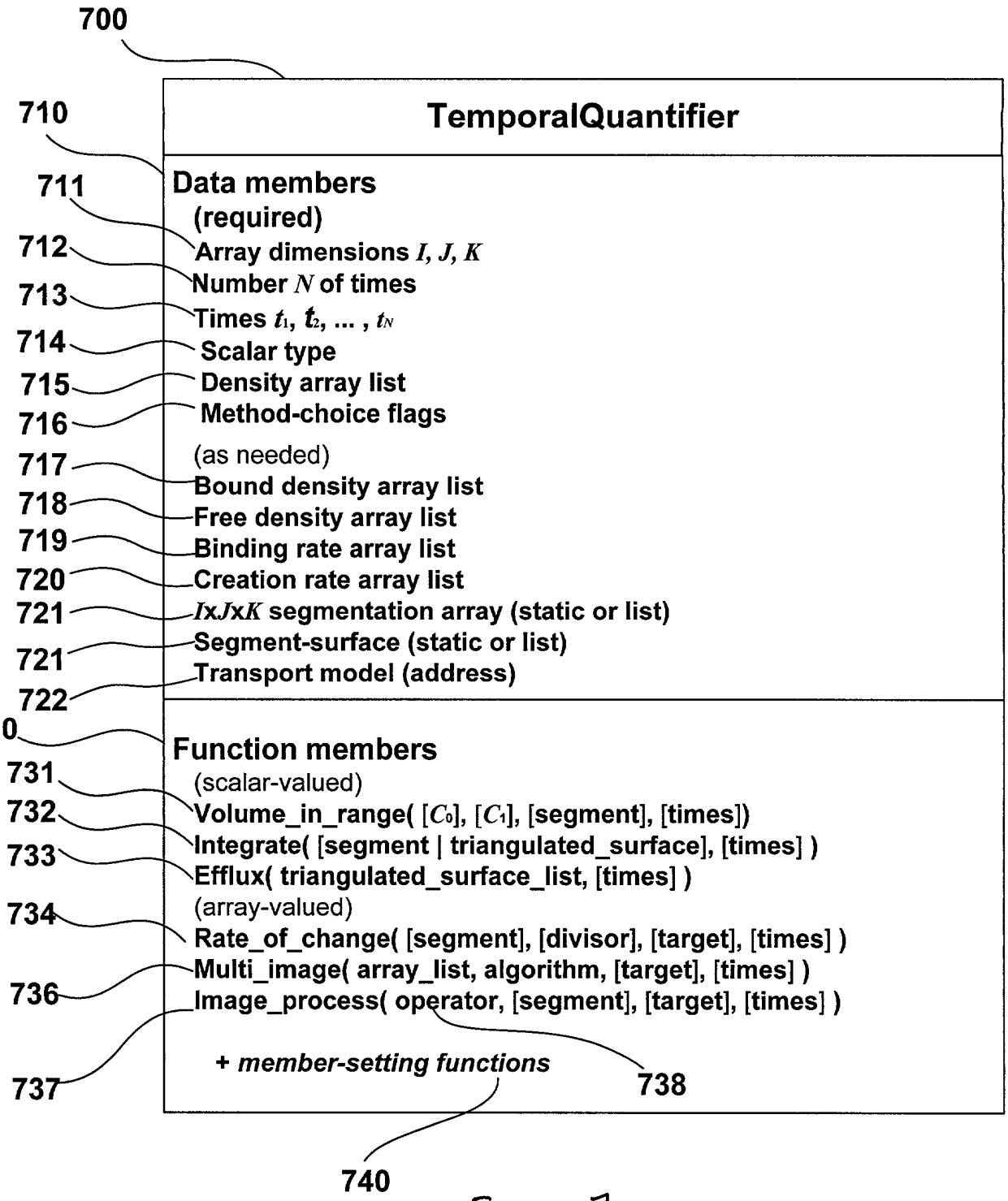

700
TemporalQuantifier
710
Data members
(required)
711
Array dimensions I, J, K
712
Number N of times
713
Times t1, t2, ... , tN
714
Scalar type
715
Density array list
716
Method-choice flags
(as needed)
717
Bound density array list
718
Free density array list
719
Binding rate array list
720
Creation rate array list
721
IxJxK segmentation array (static or list)
721
Segment-surface (static or list)
722
Transport model (address)
730
Function members
(scalar-valued)
731
Volume_in_range([C0], [C1], [segment], [times])
732
Integrate([segment | triangulated_surface], [times])
733
Efflux(triangulated_surface_list, [times])
(array-valued)
734
Rate_of_change([segment], [divisor], [target], [times])
Multi_image(array_list, algorithm, [target], [times])
736
Image_process(operator, [segment], [target], [times])
737
+ member-setting functions
738
740

FIGURE 7

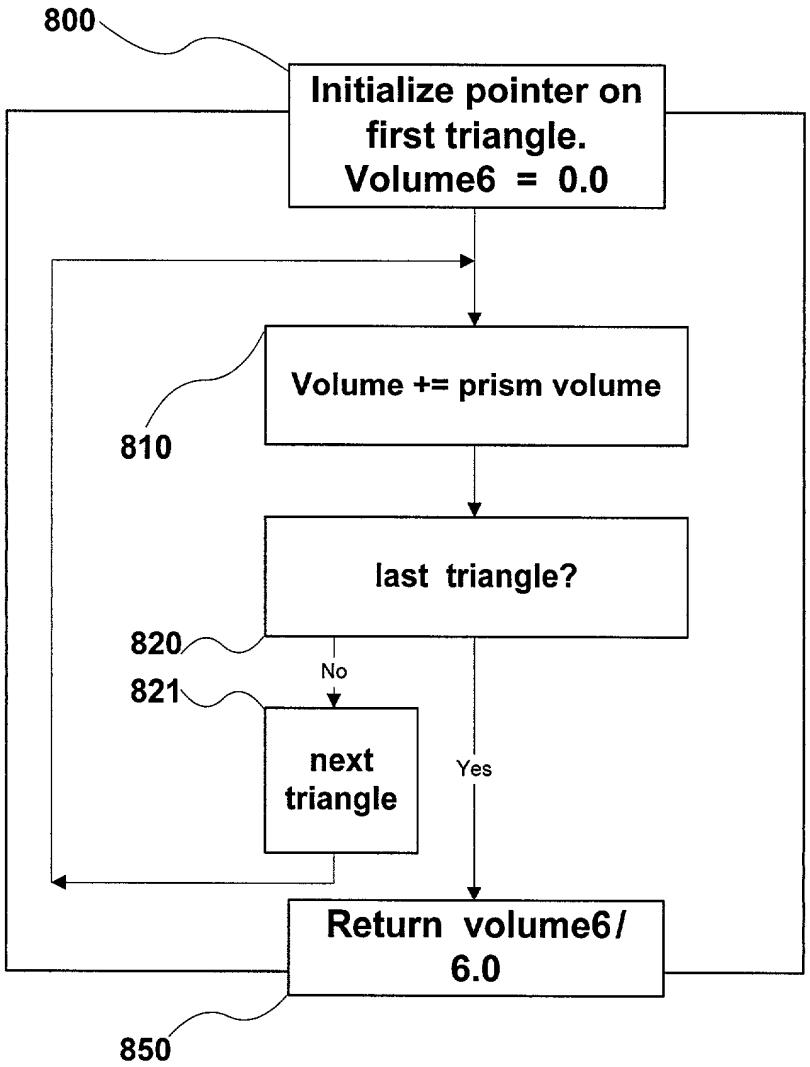

800
Initialize pointer on first triangle. Volume6 = 0.0
Volume += prism volume
810
last triangle?
820
No
821
next triangle
Yes
Return volume6/ 6.0
850

FIGURE 8

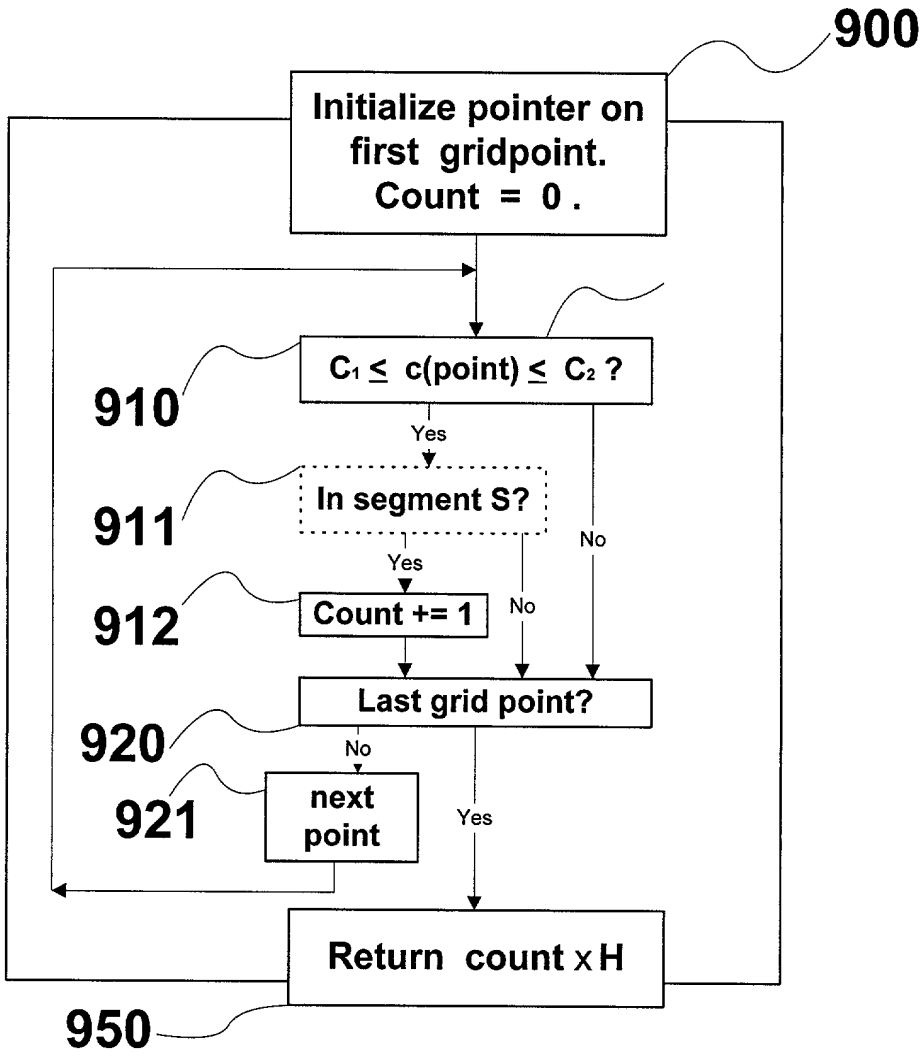

900
Initialize pointer on first gridpoint. Count = 0 .
910
C1 ≤ c(point) ≤ C2 ?
Yes
No
911
In segment S?
Yes
No
912
Count += 1
920
Last grid point?
No
Yes
921
next point
Return count x H
950

FIGURE 9

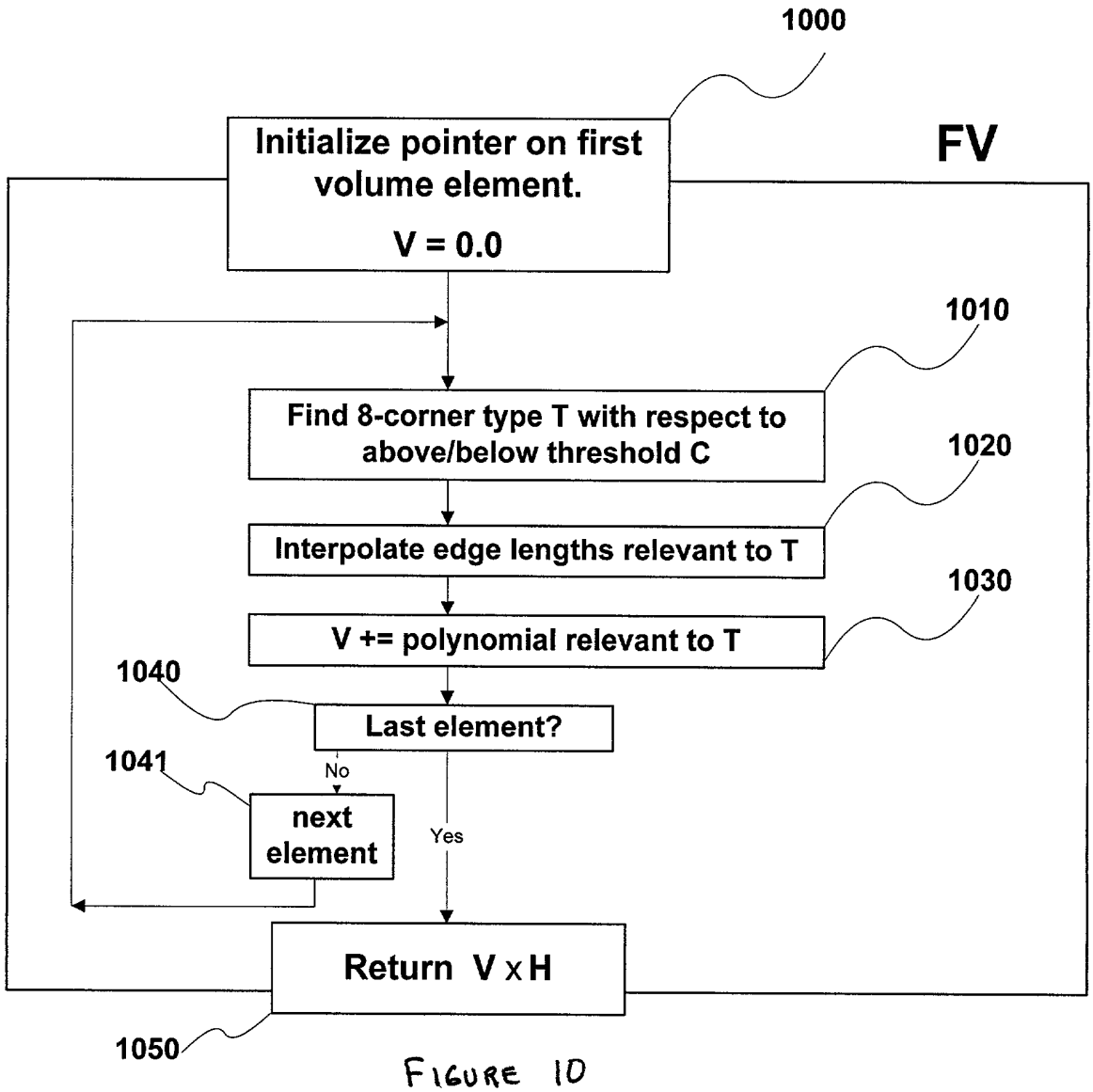

1000
FV
Initialize pointer on first volume element.
V = 0.0
1010
Find 8-corner type T with respect to above/below threshold C
1020
Interpolate edge lengths relevant to T
1030
V += polynomial relevant to T
1040
Last element?
No
1041
next element
Yes
Return V x H
1050

FIGURE 10

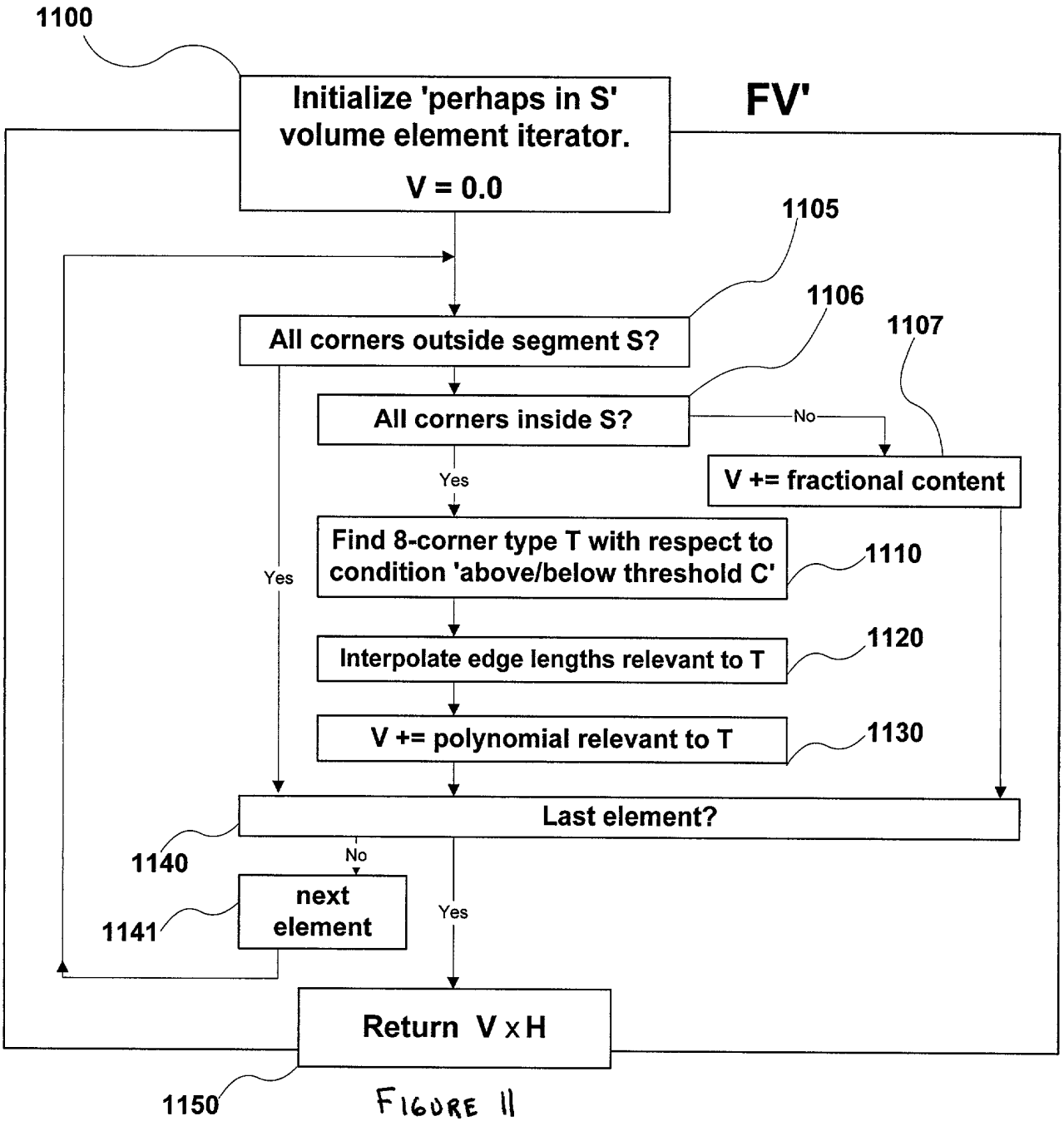

1100
Initialize 'perhaps in S' volume element iterator.
V = 0.0
FV'
1105
All corners outside segment S?
1106
All corners inside S?
1107
No
V += fractional content
Yes
Find 8-corner type T with respect to condition 'above/below threshold C'
1110
Yes
Interpolate edge lengths relevant to T
1120
V += polynomial relevant to T
1130
Last element?
1140
No
next element
1141
Yes
Return V x H
1150

FIGURE 11

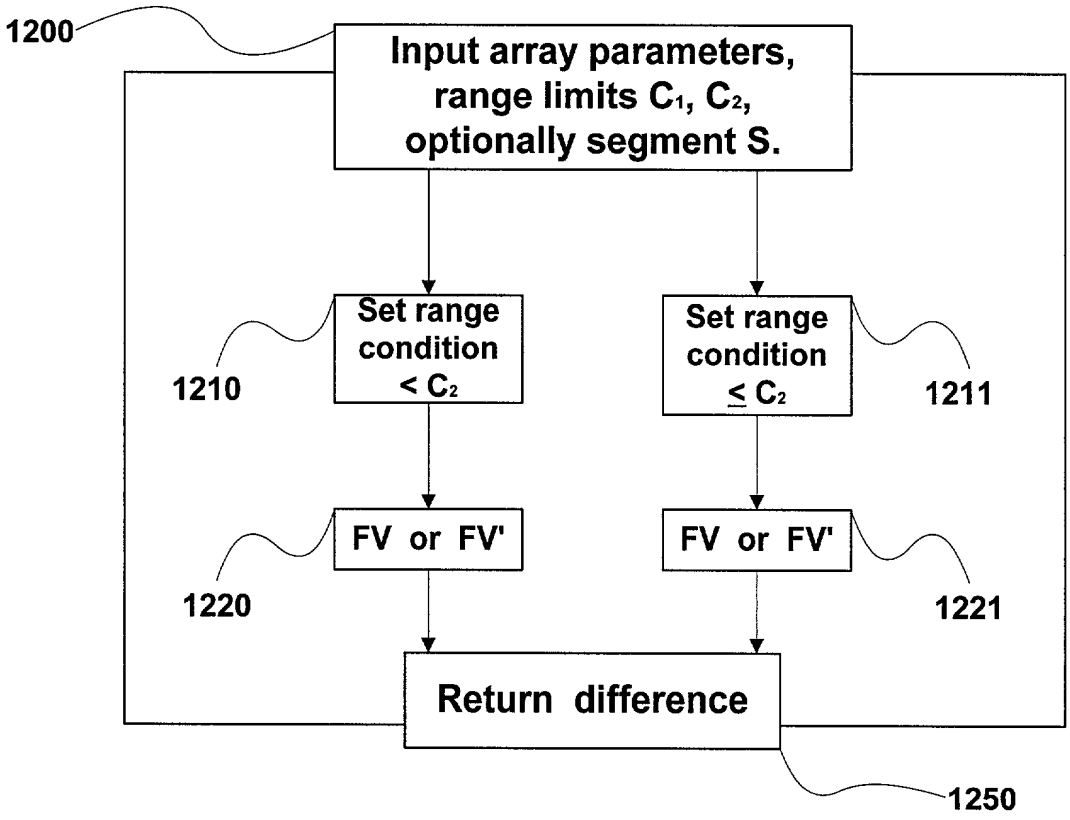

1200
Input array parameters, range limits C_1, C_2, optionally segment S.
Set range condition $< C_2$
1210
Set range condition $\leq C_2$
1211
FV or FV'
1220
FV or FV'
1221
Return difference
1250

FIGURE 12

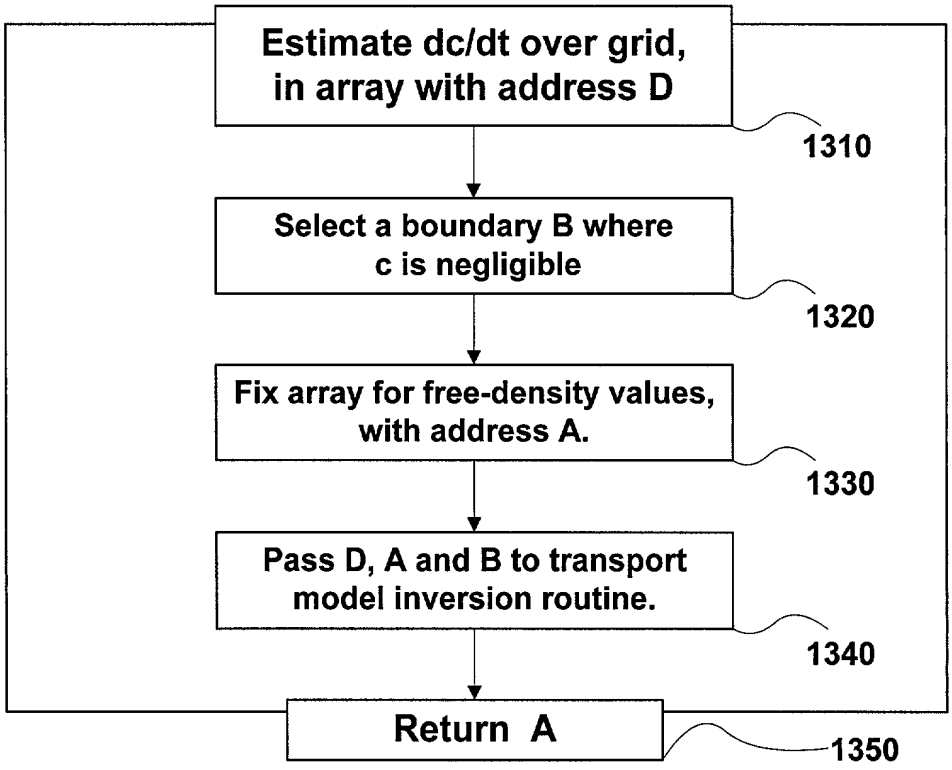

Estimate dc/dt over grid, in array with address D
1310
Select a boundary B where c is negligible
1320
Fix array for free-density values, with address A.
1330
Pass D, A and B to transport model inversion routine.
1340
Return A
1350

FIGURE 13

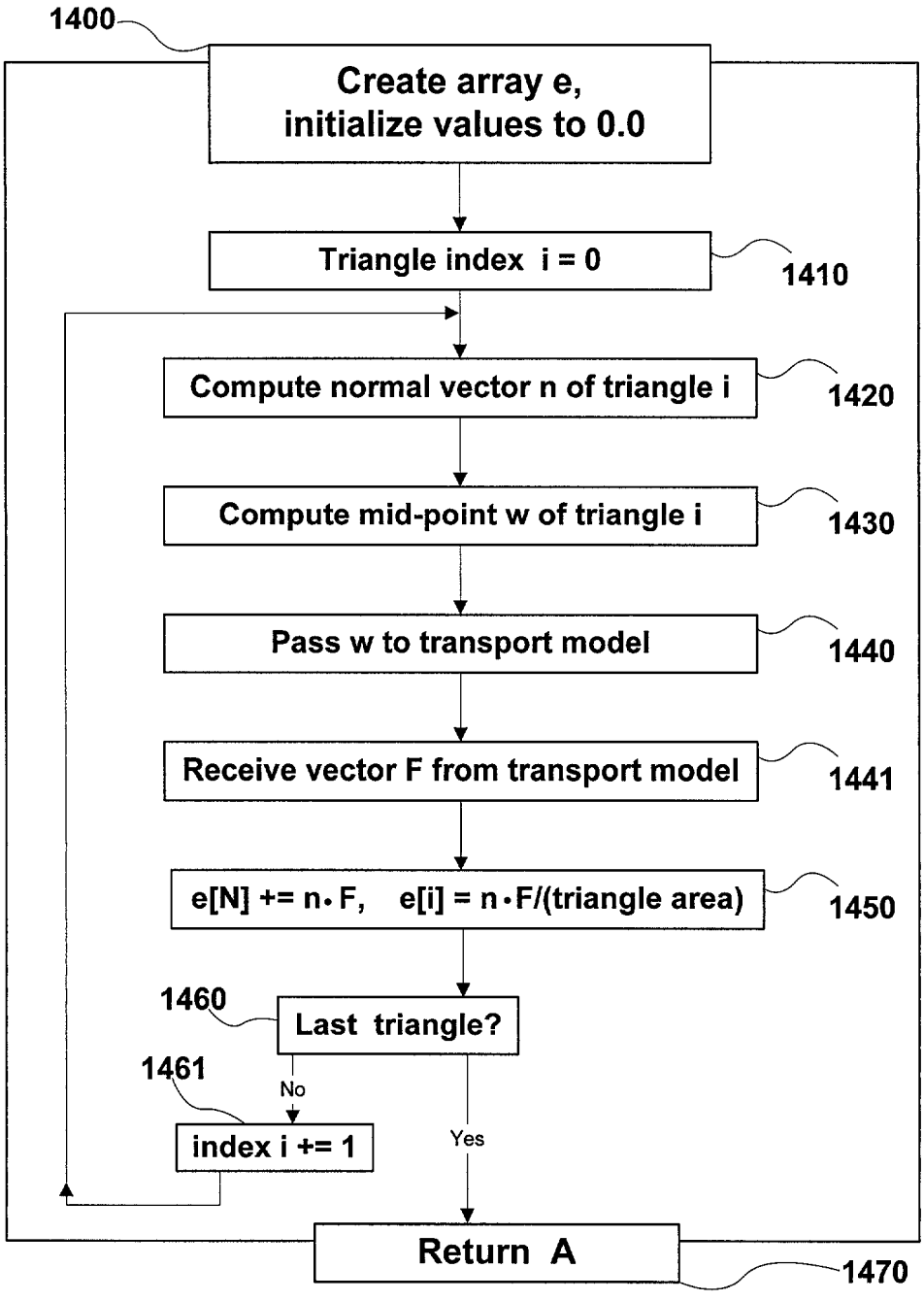

1400
Create array e, initialize values to 0.0
Triangle index i = 0
1410
Compute normal vector n of triangle i
1420
Compute mid-point w of triangle i
1430
Pass w to transport model
1440
Receive vector F from transport model
1441
e[N] += n • F, e[i] = n • F/(triangle area)
1450
1460
Last triangle?
1461
No
index i += 1
Yes
Return A
1470

FIGURE 14